

# Ritorno del calcolo vettoriale al CILEA

**Giampaolo Bottoni, Giovanni Meloni**

*CILEA, Segrate*

## *Abstract*

Il CILEA torna ad affiancare alle macchine parallele degli elaboratori vettoriali.

L'ultimo calcolatore installato è stato un Convex a due processori della serie C3800 spento nel 1997. Il nuovo elaboratore, che diventerà operativo nel mese di Gennaio, è un NEC SX5/4S a quattro processori.

*Keywords:* Supercalcolo, Algebra lineare, Calcolo parallelo, Simulazione numerica

Il CILEA ritorna al vettoriale decidendo di affiancare due tipologie di architetture (la parallela e la vettoriale) in grado di offrire all'utenza due ambienti operativi ottimizzati per le diverse applicazioni di calcolo ad alte prestazioni.

Le caratteristiche tecniche del nuovo elaboratore prevedono una architettura vettoriale-parallela in ambiente shared memory in grado di supplire a quelle limitazioni a cui una architettura seriale-parallela può essere talvolta soggetta.

Ogni processore (con potenza di picco di 4 GFLOPS) contiene una vector unit, una scalar unit con cache-on-chip ed una memory interface.

La vector unit è dotata di 144 KBytes di registri vettoriali (8 vector register e 64 vector data register da 64 bits) utilizzati come cache vettoriale ad alte prestazioni in grado di ridurre significativamente il flusso di dati verso la memoria.

Vi sono 8 vector pipeline sets ( add-shift, multiply, divide, logical) in grado di eseguire 8 operazioni in parallelo sui dati per ciclo.

Nell'unità scalare le pipeline scalari comprendono due unità floating point (addizione, moltiplicazione, divisione) e due unità interi.

La memoria è di tipo SDRAM con bandwidth di 32 gigabytes al secondo per processore.

I canali di I/O prevedono una connessione di tipo HIPPI-800 da 100 megabyte al secondo e SCSI (Ultra & FWD).

L'ambiente operativo è Unix con compilatori vettoriali FORTRAN 90 e C++. Sono inoltre disponibili tools per l'ottimizzazione ed il controllo delle prestazioni.

Saranno anche disponibili software applicativi di simulazione che traggono particolari vantaggi prestazionali in questo ambiente come : NASTRAN, PAM-CRASH, STAR-Cd, etc.

Sono previsti corsi di introduzione all'utilizzo della nuova macchina che saranno dedicati sia all'ambiente operativo ed al software di sistema sia all'ottimizzazione dei codici applicativi.

Perché il vettoriale?

La disponibilità di un ambiente che "premi" determinati tipi di algoritmi e determinati stili di programmazione favorisce la ricerca di algoritmi e metodologie che rientrino nella categoria "favorita" dall'ambiente stesso.

Gli algoritmi di interesse numerico informatico possono, a grosse linee, venire raggruppati in tre, ormai classiche, categorie.

- Sequenziali: nei quali il principale obiettivo è (a) la riduzione del numero totale di operazioni richieste e (b) la "concisione" e facilità di programmazione dell'algoritmo (sono migliori gli algoritmi che, a parità di risultato, richiedono poche istruzioni e poche operazioni)
- A parallelismo fine ossia tali da richiedere una singola operazione aritmetica

applicata ad una grossa molteplicità di dati. Sono migliori gli algoritmi che sanno evidenziare tale molteplicità di dati anche a costo di un limitato aumento del numero di operazioni da fare (esistono casi limiti in cui anche un raddoppio del numero di operazioni da fare rispetto ad un approccio sequenziale è ancora considerato accettabile).

- A parallelismo di compiti, in cui esistono fasi lavorative separate e non interferenti l'una con l'altra. Sono dunque migliori gli algoritmi che riducono il numero di punti di sincronizzazione tra le varie fasi anche a costo di una ripetizione (solo apparentemente irrazionale) di determinati calcoli a patto che questo consenta una forte diminuzione dello scambio di informazioni tra lavori effettuati in contemporanea e dunque riduca la comunicazione tra i lavori contemporaneamente in corso.

L'evoluzione storica dell'informatica ha privilegiato in ordine cronologico tutte e tre queste categorie di algoritmi.

Agli albori dell'informatica, quando era costoso tanto la singola operazione quanto l'estensione stessa del programma oltre che quella dei dati, lo sforzo è stato rivolto alla individuazione di algoritmi esprimibili con pochissime istruzioni e tali da richiedere l'uso di pochi dati ausiliari.

Con l'avvento dei primi Cray divennero di attualità algoritmi specifici per il calcolo vettoriale, privi, per quanto possibile, di salti condizionali che, per la loro imprevedibilità, impedissero la migrazione ordinata di grosse quantità di dati residenti in zone contigue della memoria verso l'unica CPU dotata di apposite unità funzionali (i registri vettoriali) e viceversa.

Con l'avvento dei transistori ad effetto di campo ed il calo vertiginoso del costo dei circuiti stampati, ha infine preso piede (ed è un effetto tuttora in corso di espansione) il calcolo parallelo a macrocompiti in cui diventa economico disporre di molte unità di calcolo adibite a compiti sia totalmente diversificati sia anche analoghi ma applicati a grossi blocchi di dati distinti e scarsamente intercorrelati.

L'evoluzione della informatica verso una sempre maggiore distribuzione del calcolo tra processori sempre più numerosi non ha tuttavia cancellato la ragione d'essere dei diversi tipi di ottimizzazione degli algoritmi.

In questo ambito l'ottimizzazione scalare è ritornata di attualità in edizione riveduta e corretta: non si tratta più di fare i conti semplicemente con le risorse di memoria totale ma con le risorse di memoria veloce (la cache) che continua ad essere limitata per motivi di economia hardware.

*E resta, o dovrebbe restare, di attualità l'ottimizzazione vettoriale se è vero, come è vero, che certi tipi di calcoli, intrinsecamente interconnessi e poco o nulla disaccoppiabili in grossi lavori indipendenti, possono essere velocizzati solo processando grosse quantità di dati tramite poche e specializzate unità funzionali.*

Se ammettiamo dunque che una grossa categoria di operazioni di algebra lineare applicata in ampi settori dell'ingegneria e della ricerca diventa sfruttabile e poco onerosa solo disponendo di calcolatori vettoriali, dobbiamo anche ammettere che è importante non solo la consapevolezza "teorica" dell'esistenza di questi strumenti di calcolo (so che nel mondo ci sono e converrà sempre farli) ma anche la disponibilità "concreta" di macchine capaci di premiare "concretamente" l'impegno allo sviluppo di tali particolari classi di algoritmi.